IN THE SPECIFICATION

Please amend the paragraph beginning on page 12, line 14, as follows:

This leads to the structure 53 as shown in Figure 5 wherein the team produces in three parallel workframe products DEVA, DEVB, DEVC product elements. This work is then the subject of a test operation, i.e., in a LAN directory TEST, the result of which is the final product, i.e., in a LAN directory PROD. Such directories are herein also called groups. A basic rule is that only in the bottom groups, modification must be done. As shown in Figure 5, product elements can only be moved to PROD 54 from TEST 55, and elements can only be moved to TEST 55 from DEVA, DEVB or DEVC. Edits can only be done through DEVA, DEVB and DEVC. The blocks 54-58 represent data sets stored in the database or in the memories 14 of the client workstations.

Please amend the paragraph beginning on page 39, line 23, as follows:

Figure 6 shows the logical element hierarchy and data flow 60 of a complex project related to a first and a second version of a product. Each block in Figure 6 represents a data unit. The development of a product starts on the low level in groups DEVA, DEVB and TEAM. These groups represent the editable data units where elements may be created, modified or updated. Each of the units DEVA and DEVB is assigned to a single user, while unit TEAM is assigned to a team of users. The next higher blocks DEVTA, DEVTB and TEAMT are assigned to the test of the elements from DEVA, DEVB and TEAM. No modifications of the product elements are permitted in the test level. The next higher levels represent a function test FTEST and a system test STEST of the second version of the product resulting from the elements treated and tested on the lower levels. At this point of time, the first version of the product PROD1 is already completed and shipped to customers. Errors which are found after the shipment are fixed and the subject of a test FIXTEST1 and followed by an integration into corrected product FIXPROD1.

Please amend the paragraph beginning on page 44, line 15, as follows:

Figure 10 shows the logical element hierarchy and workflow of a complex project which is produced by a distributed development by means of distributed computer systems 1, 2 and 3 designated in Figure 10 by blocks 100, 101 and 102. As already described in connection with Figure

3, these systems may be located a far distance from each other. Each of the systems 100, 101 and 102 comprises a first server 20 including an XSCML processor 22 and a database 24 and preferably a second server ~~26~~ 27 (Figure 2) for supporting build operations. System 102 comprises the edit section with the units DEVA, DEVB and TEAM which correspond to DEVA, DEVB and TEAM in Figure 6. Furthermore, system 102 comprises the edit test section with the units DEVTA, DEVTA and TEAMT which correspond to DEVA, DEVB and TEAMT in Figure 6. System 102 also performs the test of fixes by FIXTEST1 of the first version of the product which corresponds to FIXTEST1 in Figure 6. System 101 performs the functions test FTEST and the system test STEST of the second version of the same product corresponding to the FTEST and STEST in Figure 6. In system 100, the product versions PROD1 and PROD2 are administrated according to PROD1 and PROD2 in Figure 6. System 100 also performs the test of fixes FIXPROD1 of the first version according to FIXPROD1. The data flow in the systems 100, 101 and 102 corresponds to the data flow in the system shown in Figure 6. In each of these systems, the data flow and the processes are controlled by the local XSCML processor.

Please amend the paragraph beginning on page 45, line 5, as follows:

As shown in Figure 11, a project model 110 defined in the XSCML syntax is loaded into a master system 111 which may be a separate system or one of the systems 112, 113 and 114. As a first step, the XSCML processor 22 of the master system 111 performs a parsing and validating of the model loaded and checks the model on integrity. Copies of the model are~~than~~ then transferred through connections 115 to the local database 24 of each of the systems 112, 113 and 114 where the model is locked. The connections 115 may be established via a communication network 34 such as the Internet or an Intranet. The locked models in the databases of the systems 112, 113 and 114 are then activated by commands on connections 116 which also may include the communication network to synchronize the systems with a new or updated model with a fallback possibility.

Please amend the paragraph beginning on page 46, line 12, as follows:

Step 124 activates the stored system view. The initiating client requests the XSCML processor of the master system to activate the project definitions stored there. The activation includes a preparation of the physical environment as defined in the project definition. The process may involve the creation of files and databases and an update of the security system related to them. For the distributed systems, the master system initiates the activation of the involved distributed systems which will prepare their environment as defined in their stored and locked project definitions. If all systems indicate the successful activation, the lock ~~is taken~~ of the project definitions is taken away. The local client is then enabled to work on the project as defined in the project definitions which work may involve to manage the project definitions in step 125 and to perform user client actions by step 126.

Please amend the paragraph beginning on page 46, line 23, as follows:

Step 125 is shown in more detail in Figure 13. It includes a system view update step ~~139~~ 131 to implement changes of an existing project definition by a client request to retrieve from the XSCML processor of the master system the current project definition. That XSCML processor returns this project definition in the tag format and locks it in the update mode. The client may then do all actions provided by the system view definition step 121. Step 131 is followed by a new validation step 132 as described above after which a refresh system view step 133 is performed which sends the validated updates of the project definition to the XSCML processor of the master system. The XSCML master processor ensures that the changed project definition is loaded and activated in the same way described above for the steps 123 and 124. Dependent on the capability of the XSCML master processor, a delta format of the project definition may be kept to allow a reproduction of the project definition before it was updated. Further steps shown in Figure 13 include step 134 for deactivation of a system view so that client actions can temporarily not be performed, and step 135 for a complete removal of a system view so that it can not be used anymore. Step 125 also includes the definition of an alternate system view by step 136 which is followed by steps 137, 138 and 139 to validate, load and activate the alternate system view. Step 140 brings the control back to the calling client.